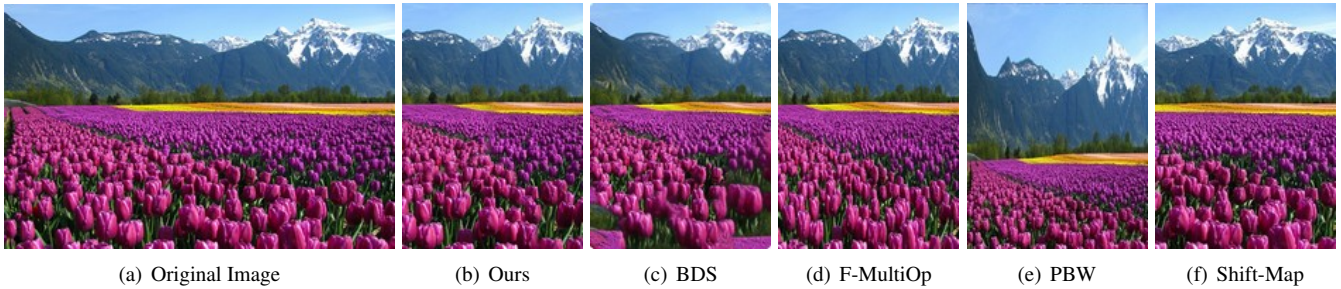


# Image Retargeting by Content-Aware Synthesis

Weiming Dong<sup>1\*</sup> Fuzhang Wu<sup>1</sup> Yan Kong<sup>1</sup> Xing Mei<sup>1</sup> Tong-Yee Lee<sup>2</sup> Xiaopeng Zhang<sup>1</sup>  
<sup>1</sup>LIAMA-NLPR, Institute of Automation, Chinese Academy of Science <sup>2</sup>National Cheng Kung University



**Figure 1:** There are large textural regions in the original image (a). By using content-aware synthesis, our method (b) can retain the texels without over-smoothing/content discontinuity (c), uneven distortion (d), over-squeezing/over-stretching (e) or layout information loss (f). Input resolution  $555 \times 347$ . Target resolution  $256 \times 347$ . 65.45% users favour our result.

## Abstract

Real-world images usually contain vivid contents and rich textural details, which will complicate the manipulation on them. In this paper, we present a content-aware synthesis method to enhance content-aware image retargeting. By detecting the textural regions in an image, the image content can be *synthesized* rather than simply distorted or cropped. This method enables the manipulation of textural & non-textural regions with different strategy since they have different natures. We propose to resize the textural regions by content-aware synthesis and non-textural regions by fast multi-operators. To achieve practical resizing applications for general images, we develop an automatic and fast texture detection method that can detect multiple disjoint textural regions. We adjust the saliency of the image according to the features of the textural regions. To validate the proposed method, comparisons with state-of-the-art resizing techniques and a user study were conducted. Convincing visual results are shown to demonstrate the effectiveness of the proposed method.

**Keywords:** Image retargeting, texture detection, texture saliency detection, content-aware synthesis

## 1 Introduction

Image retargeting has retained in the front rank of most widely-used digital media processing techniques for a long time. To adapt raw image material for a specific use, there is often the needs of achieving a target resolution by reducing or inserting image content. To protect certain important areas, some methods [Avidan and Shamir 2007; Wang et al. 2008; Panozzo et al. 2012; Lin et al. 2013] used significance maps based on local low-level features such as gradient, dominant colors, and entropy. However, high-level semantics also play an important role in human’s image perception, so usually it is necessary to better understand the content of an image to help to choose a more feasible scheme for retargeting operation. Moreover, as found in [Rubinstein et al. 2010], viewers are more sensitive to deformation than to image area loss. Therefore for some examples it is better to summarize the content rather than distort/warp or crop the origin images [Simakov et al. 2008; Wu et al. 2010; Dong et al. 2014].

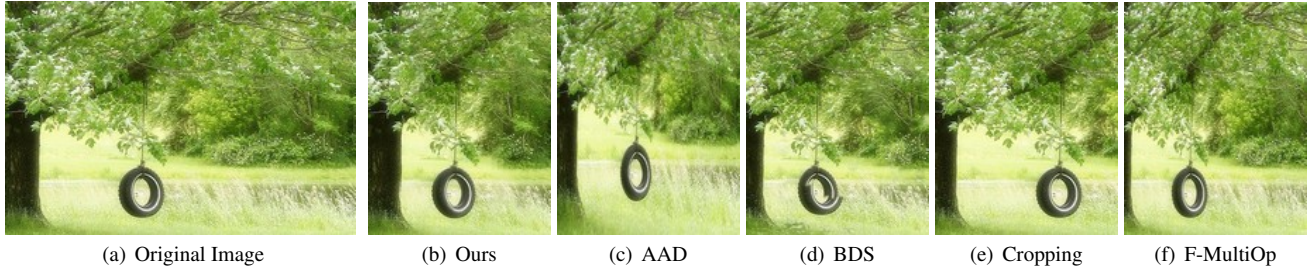
Although many retargeting methods have been proposed, a few noticeable and critically influencing issues still endure. They are mostly related to complexity of textural patterns in many natural images. Previous retargeting techniques attempt to modify the image without preserving the textural structure, and may easily result in apparent visual artifacts, such as the over-squeezing/over-stretching (Figures 1(e), 2(c)), uneven distortion (Figures 1(d), Figures 2(f), 2(f)), over-smoothing (Figures 1(c), 2(d)), scene layout information loss (Figure 1(f)), or local discontinuity (Figures 1(c), 2(d), 2(e)). The examples in Figures 1 and 2 are not special and exhibit one common problem - that is, *when images contain large textural patterns, retargeting quality could be generally affected by their complexity*. Since regularity is an important high-level feature for human texture perception [Rao and Lohse 1993] and texture exists in many natural images, this problem cannot be ignored.

We propose a novel content-aware synthesis algorithm to address textural patterns in image retargeting. In preprocessing, the textural regions (T-regions) of the input image are automatically detected based on local variation measures and each pixel in a T-region is assigned a significance value. In the retargeting process, the input image is first retargeted to the target size by fast multi-operators (F-MultiOp). Then, the T-regions are regenerated by *synthesis*, which arranges sample patches with respect to the neighborhood metric and patch position information (Figures 1(b), 2(b)). The patches with higher significance values have higher probability to appear in the result. With the content-based information and texture synthesis technique, the proposed approach can better protect both the local shape of the texture elements (texels) and the global visual appearance of the T-regions than previous image retargeting methods.

Compared with recent studies on image retargeting, the major contributions of the proposed approach are as follows:

- A fast and automatic method to detect the T-regions in an image. This process makes it possible for retargeting operation to treat the T-regions and NT-regions (non-textural regions) with different schemes.
- A novel texture saliency detection method to generate significance map in a T-region, which is based on both color and texture features.
- A synthesis-enhanced image retargeting approach is proposed to ease unpleasant visual distortions caused by warping or

\*e-mail: Weiming.Dong@ia.ac.cn



**Figure 2:** Almost all the contents of (a) are textures. (c) AAD is generated by [Panozzo et al. 2012]. 63.64% users favour our result.

scaling to overall texels in T-regions. Thus, our approach can yield better results in terms of texture element (texel) shape and preservation of globally varying effect compared with related approaches.

The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 presents the proposed approaches. Section 4 discusses the experimental result and Section 5 presents the conclusion and future work.

## 2 Related Works

### 2.1 Image Retargeting

Numerous content-aware image retargeting techniques have recently been proposed. Cropping has been widely used to eliminate the unimportant information from the image periphery or improve the overall composition [Zhang et al. 2013; Yan et al. 2013a; Zhang et al. 2014]. Seam carving methods iteratively remove a seam in the input image to preserve visually salient content [Avidan and Shamir 2007; Rubinstein et al. 2008]. A seam is a continuous path with minimum significance. Multi-operator algorithms combine seam carving, homogeneous scaling and cropping to optimally resize images [Rubinstein et al. 2009; Dong et al. 2009; Dong et al. 2012]. Pritch et al. [2009] introduced Shift-Map that removed or added band regions instead of scaling or stretching images. For many cases these *discrete* approaches can generate pleasing results, however, the seam removal may cause discontinuous artifacts, and cropping is unsuitable for the case when there are visually salient contents near the borders of images.

*Continuous* retargeting methods have been realized through image warping or mapping by using several deformation and smoothness constraints [Gal et al. 2006; Wolf et al. 2007; Wang et al. 2008; Zhang et al. 2008; Krähenbühl et al. 2009; Guo et al. 2009]. A finite element method has also been used to formulate image warping [Kaufmann et al. 2013]. Recent continuous retargeting methods focus on preserving local structures. Panozzo et al. [2012] minimize warping energy in the space of axis-aligned deformations to avoid harmful distortions. Chang et al. [2012] couple mesh deformations with similarity transforms for line features to preserve line structure properties. Lin et al. [2013] present a patch-based scheme with an extended significance measurement to preserve shapes of both visual salient objects and structural lines. These approaches perform well on shape preservation of salient objects but will easily cause distortion to all the texels of a T-region.

*Summarization*-based retargeting approaches eliminate repetitive patches instead of individual pixels and preserve patch coherence between the source and target image during retargeting [Simakov et al. 2008; Cho et al. 2008; Barnes et al. 2009]. These techniques measure patch similarity and select patch arrangements that fit together well to change the size of an image. However, due to the lack of enough content information, the major drawback of such

methods is that the global visual effect may be discarded and some regions may be over-smoothed when the target size is small.

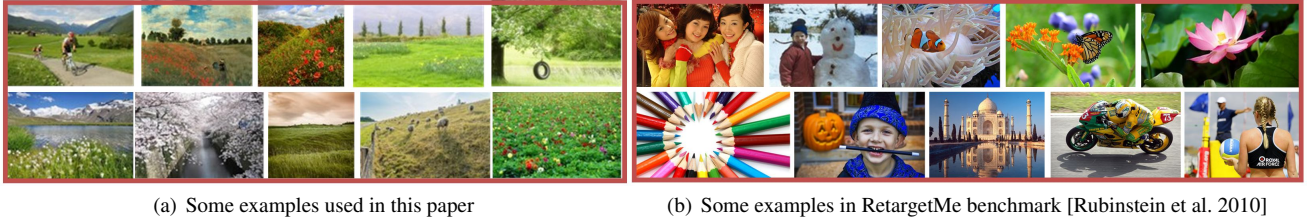
High level image content informations are analyzed and integrated in some recent summarization approaches. For example, Wu et al. [2010] detect the corresponding lattice of a symmetry image region and retarget it by trimming the lattice. Basha et al. [2013] employ depth information to maintain geometric consistence when retargeting stereo images by seam carving. Dong et al. [2014] detect similar objects in the input image and then use object carving to achieve a natural retargeting effects with minimum object saliency damage. There also exist a few temps to deal with textures for better retargeting. Kim and Kim [2011] exploit the higher order statistics of the diffusion space to define a reliable image importance map, which can better preserve the salient object when it is located in front of a textural background. This approach does not consider how to preserve the visual effects of textural regions. Zhang and Kuo [2012] resize the salient and irregular regions by warping and re-synthesize the regular regions. However, in [Zhang and Kuo 2012] the authors did not address what situations the regularity detection algorithm works which questions its robustness. On the other hand, the synthesis algorithm they used can only deal with isotropic textures which is not fit for most natural images with vivid anisotropic texture regions.

### 2.2 Texture Detection and Synthesis

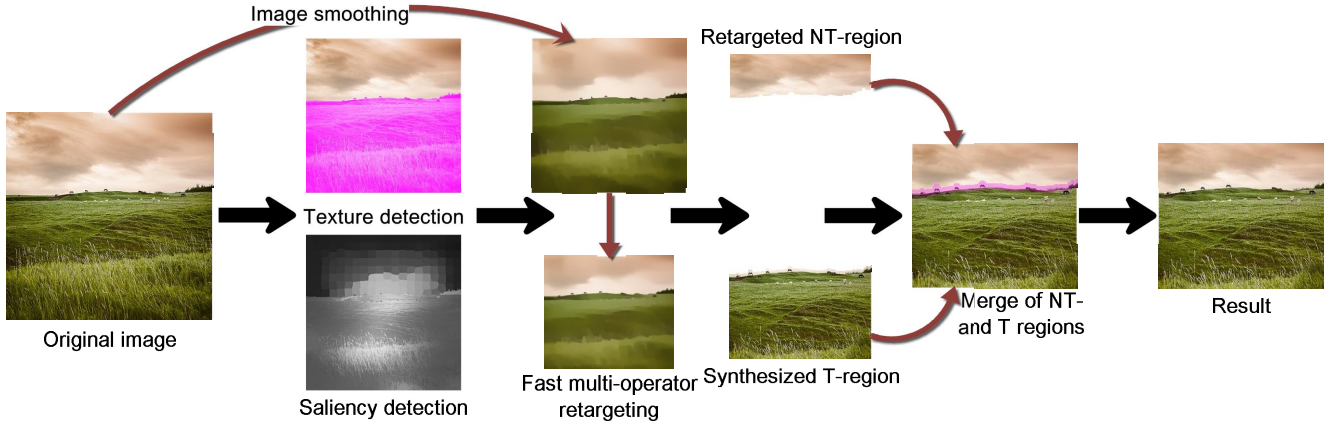
The adaptive integration of the color and texture attributes in image segmentation is one of the most investigated topics of research in computer vision (surveyed in [Ilea and Whelan 2011]). However, most of the image segmentation algorithm do not clearly illustrate the type of each region (textural or non-textural) in the result. Targhi et al. [2006] present a fast texture descriptor based on LU transform, but how to determine if a pixel is texture or non-texture according to the feature values is not discussed. Bergman et al. [2007] present an intuitive texture detection method which is based on contrast and disorganization measurements of image blocks. The method is not effective on noisy images which tends to have decreasing contrast and often generate many disjoint areas. Todorovic and Ahuja [2009] formulate the detection of texture subimages as identifying modes of the pdf of region descriptors. However, the method is not efficient (5 minutes for a  $512 \times 512$  image) for practical image retargeting applications.

Texture synthesis is a general example-based methodology for synthesizing similar phenomena [Wei et al. 2009]. However, the basic MRF-based scheme in most existing texture synthesis methods cannot adequately handle the globally visual variation of texels, such as perspective appearance and semantic content distribution. Dong et al. [2008] present a perspective-aware texture synthesis algorithm by analyzing the size variation of the texel, but verbatim copying artifacts also often appear in their results. On the other hand, common texture synthesis algorithms are designed for enlargement and can not be directly used for image retargeting applications. Wei [2008]





**Figure 3:** Different from most of the images in the RetargetMe benchmark, the images used in this paper all contain large textural regions.



**Figure 4:** Framework of our method. Input resolution  $400 \times 400$ . Target resolution  $240 \times 210$ . 47.27% users favour our result.

presents inverse texture synthesis approach to generate a smaller example from a large input texture. However, for globally varying textures, the output quality of this approach usually depends on the accuracy of the original map. Therefore, if applied to normal T-region retargeting, it will easily lose the globally visual variation or damage the local content continuity of the original image in the result.

### 3 Retargeting by Synthesis

#### 3.1 System Overview

Some standard examples studied in our work are shown in Figure 3(a). Different from most examples in the RetargetMe benchmark, our images all contain one or more large textural regions, which bring new challenges to image retargeting. Figure 4 illustrates the framework of the proposed method which consists of two main steps: *preprocessing* and *image retargeting*. In the preprocessing step, the input image is segmented into one or more T-regions and one NT-region (we treat disjoint NT-regions also as one region) by texture detection (Sec. 3.2). A hierarchical saliency detection for texture is then performed to generate a significance map for each T-region (Sec. 3.3). The significance map of the whole image is also adaptively adjusted according to the percentage of areas of T-regions. Afterwards, the input image is filtered by structure-preserving image smoothing. In the image retargeting step, fast multi-operator (F-MultiOp) [Dong et al. 2012] method is firstly used to resize the filtered input image to the target size (Sec. 3.4). The process of resizing the smoothed image is used to guide the resizing process of the original image in order to eliminate the effect of textural details (Sec. 3.4). We then re-generate the T-regions of the resulting image via the proposed content-aware synthesis operator, in order to maintain the perspective variation, content diversity, as well as the texel shapes (Sec. 3.5). Finally, we refine the boundaries between T- and NT-regions by re-synthesizing the pixels of

the boundary areas (Sec. 3.6).

#### 3.2 Automatic Texture Detection

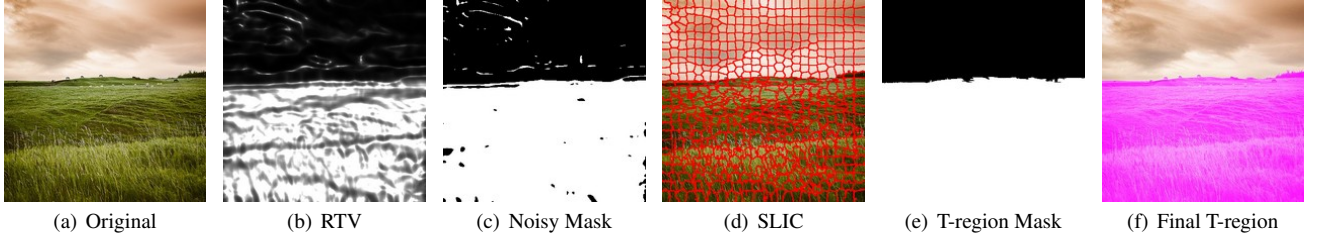
The first step for our image retargeting system is to locate the T-regions. Recently, local variation measures were used to smooth texture and extract structures from an image [Buades et al. 2010; Xu et al. 2012; Karacan et al. 2013]. However, this kind of approaches all can not provide the positional information of textures, especially for most natural images which contain both T- and NT-regions. We develop a fast texture detection method based on the measure of relative total variation (RTV) [Xu et al. 2012]. Given an input image, we first calculate the *windowed total variations*  $\mathcal{D}_x(p)$  (in the  $x$  direction) and  $\mathcal{D}_y(p)$  (in the  $y$  direction) for pixel  $p$ . *Windowed inherent variations*  $\mathcal{L}_x(p)$  and  $\mathcal{L}_y(p)$  are also calculated. Details of calculating the windowed variations are described in [Xu et al. 2012]. We then calculate the reliability of pixel  $p$  being a texture pixel as [Xu et al. 2012]:

$$R(p) = \frac{\mathcal{D}_x(p)}{\mathcal{L}_x(p) + \epsilon} + \frac{\mathcal{D}_y(p)}{\mathcal{L}_y(p) + \epsilon}, \quad (1)$$

where the division is an element wise operation.  $\epsilon = 10^{-5}$  is used to avoid division by zero.

After calculating the reliability of each pixel, we use an iterative algorithm to set a threshold  $R_T$  to determine the textural pixels. We first calculate the average reliability  $R_A$  of all the pixels and use  $R_T = R_A$  to separate the pixels into two parts. The pixels which  $R(p) \geq R_T$  are set as textural pixels (T-pixels) and  $R(p) < R_T$  as non-textural pixels (NT-pixel). We then calculate the average reliability of T-pixels as  $R_A^T$  and the one of NT-pixels as  $R_A^{NT}$ . The new threshold calculated as  $R'_T = \alpha \cdot R_A^T + (1.0 - \alpha) \cdot R_A^{NT}$ , where  $\alpha = 0.5$  in all our experiments. We update  $R_T = R'_T$  and repeat the above steps until  $|R'_T - R_T| < \epsilon$ .

As shown in Figure 5(c), we can get a noisy texture mask after



**Figure 5:** Texture detection. (e) Mask refined by local convex hull and superpixels. The final T-region boundary is refined by using ANN.

segmenting the original image into T-pixels and NT-pixels. This texture mask is then improved by an iterative procedure (see Figure 5(d)). A pixel is labeled as texture if it falls in the convex hull of its neighboring pixels labeled as clear. The step repeats until convergence. Then an image is oversegmented into superpixels [Achanta et al. 2012]. A super-pixel is labeled as texture if more than half of its pixels are labeled as texture. The texture mask can be further improved by KNN matting [Chen et al. 2013] in order to get more accurate boundaries for T-regions. As shown in Figure 5(f), our algorithm can accurately detect the grassland as a T-region. Please see more texture detection results and the analysis of the accuracy of the algorithm in the supplemental material.

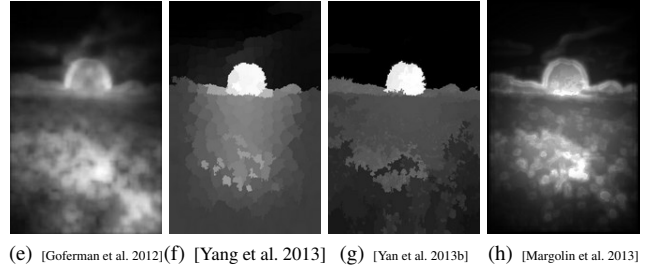
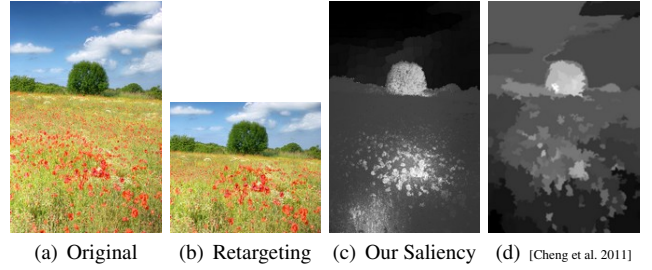
### 3.3 Texture-Based Significance Map Generation

Pixel significance measurements have been commonly used in image retargeting approaches. Usually saliency map is employed to help generate the significance map of the input image. However, the purpose of almost all of current saliency detection algorithms is to detect and segment the distinct salient objects. *To the best of our knowledge, a saliency detection algorithm aiming at marking the visually important areas (not objects) of a texture has not been proposed.* We present a hierarchical framework to deal with saliency detection of a texture. We first segment the image into  $M$  patches by using the SLIC method [Achanta et al. 2012]. Since a texture usually does not contain a distinct salient object, the saliency detection becomes determining the patches which are visually unique from others. Previous approaches usually use color or contrast information to evaluate the visual difference between pixels or patches [Cheng et al. 2011; Margolin et al. 2013], but this is not effective enough for dealing with texture images, as shown in Figure 6. In our approach, in order to better evaluate the saliency of a texture, we integrate 2D Gabor filter [Pang et al. 2013] with 4 frequencies and 6 directions to extract the texture features of the T-regions. For each SLIC patch  $A_i$ , we calculate the average and variance of Gabor values of all the pixels in it and then get a 48D texture feature. Thus, we define the visual uniqueness saliency cue of  $A_i$  as a weighted sum of color difference and texture difference from other patches:

$$U_i = \sum_{j=1}^M w(A_i) \exp\left(\frac{-D_s(A_i, A_j)}{\sigma_s^2}\right) (\|\mathbf{C}_i - \mathbf{C}_j\|^2 + \|\mathbf{G}_i - \mathbf{G}_j\|^2),$$

where  $\mathbf{C}_i/\mathbf{C}_j$  is the average color of a patch,  $\mathbf{G}_i/\mathbf{G}_j$  is the texture feature. The color feature and texture feature are both normalized to  $[0, 1]$ .  $w(A_i)$  counts the number of T-pixels in  $A_i$ . Patches with more T-pixels contribute higher visual uniqueness weights than those containing less T-pixels.  $D_s(A_i, A_j)$  is the square of Euclidean distance between patch centroids of  $A_i$  and  $A_j$ , and  $\sigma_s$  controls the strength of spatial weighting. In our implementation, we set  $\sigma_s^2 = 0.5$  with pixel coordinates normalized to  $[0, 1]$ .

Similar as [Yan et al. 2013b], we also add the location heuristic that in many cases pixels close to a natural image center could be



**Figure 6:** Qualitative comparison of saliency detection on a normal image. Previous methods fail to detect the salient dark green grass on the left-bottom. 61.82% users favour our result.

salient:

$$H_i = \frac{1}{w(A_i)} \sum_{x_j \in A_i} \exp(-\lambda \|\mathbf{x}_j - \mathbf{x}_c\|^2),$$

where  $\mathbf{x}_j$  is the coordinate of a pixel in patch  $A_i$ , and  $\mathbf{x}_c$  is the coordinate of image center.  $\lambda = 9$  to balance the visual uniqueness and location cues. We combine  $H_i$  with  $U_i$  to get the saliency of patch  $A_i$ :

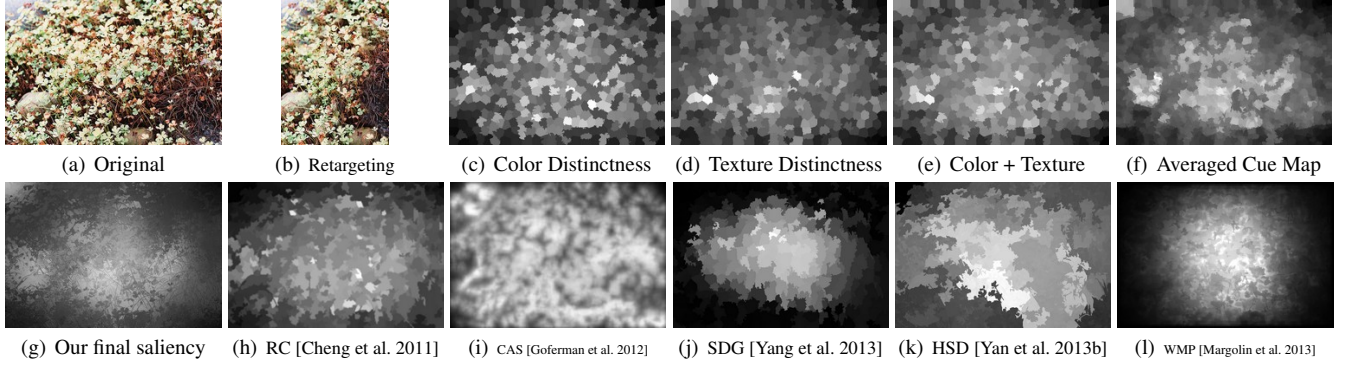
$$S_i = U_i \cdot H_i.$$

For further robustness, we compute patch-based saliency at three scales:  $M = 100, 500, 1000$  and average them pixel by pixel. As shown in Figure 7(f), we can get a coarse saliency map by using the above patch-based hierarchical method. Finally, we adopt an image up-sampling method [Criminisi et al. 2010] to refine the coarse saliency map and assign a saliency value to each image pixel. We define the saliency  $\tilde{S}_i$  of a pixel as a Gaussian weighted linear combination of the saliency of its  $N$  neighbourhoods:

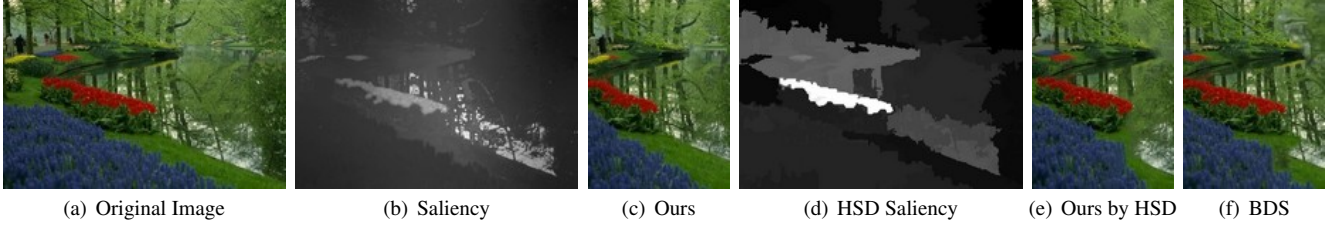
$$\tilde{S}_i = \frac{1}{Z_i} \sum_{j=1}^N \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2 + \|\mathbf{g}_i - \mathbf{g}_j\|^2 + \|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma}\right) S_j,$$

where  $\mathbf{c}_i$  is the pixel color,  $\mathbf{g}_i$  is the Gabor texture feature, and  $\mathbf{x}_i$  is the pixel coordinate. We set  $\sigma = 30$  in all our experiments. Result is shown in Figure 7(g). Similar refinement method is used

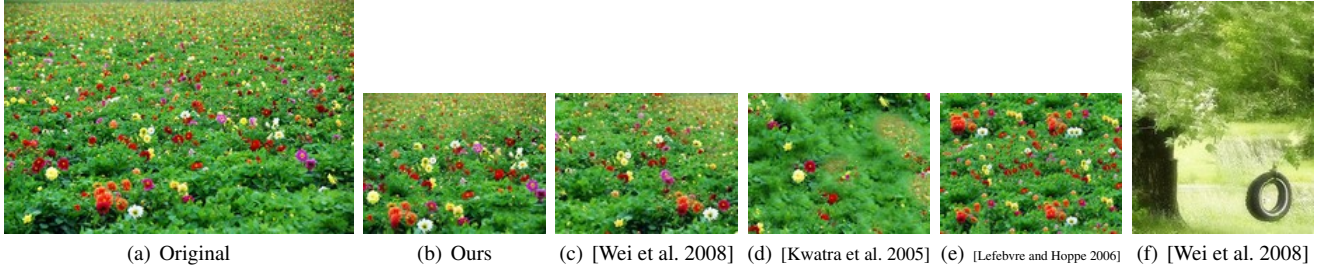




**Figure 7:** Qualitative comparison of saliency detection on a pure texture image. (c), (d), (e) are calculated at Layer 2 with  $M = 500$ .



**Figure 8:** Comparison of retargeting results by using our saliency map and by using the one of HSD method [Yan et al. 2013b]. More semantic informations (e.g. people and yellow flowers) are kept in our result while better preserving the boundary continuity of the grassland.



**Figure 9:** Comparison of previous normal texture synthesis methods to our method. Our method can both preserve the perspective effect and the salient areas in the result. (f) is the synthesis result of Figure 2(a) by using inverse texture synthesis [Wei et al. 2008].

in [Perazzi et al. 2012], but they only consider the color and position features.

As discussed above, previous saliency detection approaches are usually designed to highlight the salient object(s). As shown in Figure 7, for an image in our dataset, previous methods either over-darken or over-highlight most part of a T-region. They also have difficulties with accurately detecting the visually important areas of T-regions due to the lack of texture features. On the other hand, the content balance will be easily damaged during retargeting if the saliency values of T-regions are too smaller or too larger than NT-regions, especially when the sizes of T- and NT-regions are similar. Therefore, to address these problems, in the saliency map of the whole image, we replace the parts of T-regions with the saliency maps generated by our method. For the generation of initial saliency map, we use the method in [Yan et al. 2013b] if the area of NT-region is less than 30% of the image since this method is good at distinguishing salient objects from complex background patterns. Otherwise, we use the method in [Yang et al. 2013] to generate a more balanced initial saliency map. We use the saliency map as the significance map for retargeting operation. In Figure 8 we show an example of using different saliency maps to retarget an image. We can see that our method can highlight more visually unique contents in the saliency map than HSD method.

### 3.4 Initial Retargeting

As an initial retargeting operation, we first smooth the original image by structure extraction [Xu et al. 2012]. We then use F-MultiOp method [Dong et al. 2012] to resize the smoothed image to the target size. The significance map is utilized to preserve the important areas of both T- and NT-regions. The operation details are recorded, including the numbers of the three operators (i.e., seam carving, homogeneous scaling, and cropping) and the paths of pixels used by seam carving. Finally the original image is retargeted by copying these operations. This scheme can efficiently eliminate the unexpected affects of large-magnitude gradients of complex texture details to seam carving. After initial retargeting, the resized NT-region will be directly used in the final result, but we re-generate the T-regions by content-aware synthesis.

### 3.5 Content-Aware Synthesis

We synthesize a T-region of the resized image by using the original T-region as the example. However, for most images, directly synthesizing the content by normal texture synthesis algorithms cannot generate satisfied result or even change the semantics of the image and introduce obvious boundary discontinuity. The global visual

appearance may be damaged when the resized ratio is large. As shown in Figures 9(c)-9(f), the perspective characteristic no longer exists and the spatial structure of the content is also damaged. For our content-aware image resizing application, the synthesis algorithm should preserve the global visual appearances of the original T-regions as well as the local continuity.

**Initialization** We employ patch-based synthesis framework which is effective for image textures to synthesize the resized T-regions. In our experiments, we find that a *good* initialization will increase the quality of the resized results. Therefore, we use the resized T-regions generated by F-MultiOp in initial retargeting as the initial guess. With the help of significance map during F-MultiOp, this will effectively preserve the global visual appearance and the visually salient areas in the result.

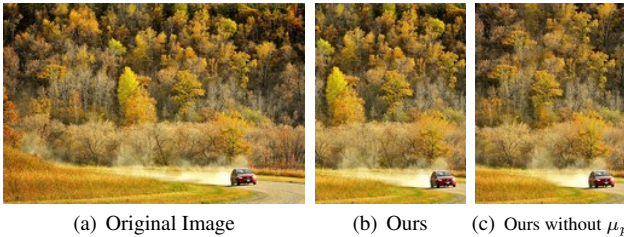
**Neighborhood metric** The neighborhood similarity metric is the core component of example-based texture synthesis algorithms [Wei et al. 2009]. We denote  $Z_p$  as the spatial neighborhood around a sample  $p$ , which is constructed by taking the union of all pixels within its spatial extent defined by a user specified neighborhood size. We formulate the distance metric between the neighborhoods of two sample  $p$  and  $q$  as:

$$M(Z_p; Z_q) = \mu_p \cdot \left( \sum_{p' \in Z_p} \|c_{p'} - c_{q'}\|^2 + \omega \cdot \|x_{p'} - x_{q'}\|^2 \right), \quad (2)$$

where  $p'$  runs through all pixels  $\in Z_p$ ,  $q' \in Z_q$  is the spatially corresponding sample of  $p'$ ,  $c$  represents the pixel color in RGB space, and  $x$  is the local coordinate of a sample pixel. Different from traditional texture synthesis that usually defines the neighborhood metric as a simple sum-of-squared of the pixel attributes (such as colors and edges), we add the spatial information to the neighborhood metric. This item can preserve the global appearance without causing over-smoothing and generating obvious partial/broken objects (detailedly discussed in Section 4). In Equation (2),  $\mu_p$  is a penalty coefficient which is used to avoid overusing the same patches in the resulting image:

$$\mu_p = 1 + \beta \cdot t_p,$$

where  $t_p$  is number of times that patch  $Z_p$  has been used in the resulting image,  $\beta = 10$  is a constant. In Figure 10 we can see the importance of adding  $\mu_q$  to the neighborhood metric in avoiding unexpected repeat patterns. Note that for Figure 10(c) we also did not integrate the significance map during the initial retargeting process, so the salient yellow trees in the middle of the original image are lost in the result.



**Figure 10:** The artifact of repeat patterns can be avoided by increasing the neighborhood matching cost with a penalty coefficient.

**Optimization** Therefore, given an original exemplar T-region  $\mathcal{I}$ , our goal is to synthesize an output  $\mathcal{O}$  that contains similar visual appearances to  $\mathcal{I}$ . We formulate this as an optimization problem via the following energy function:

$$E(\mathcal{I}; \mathcal{O}) = \sum_{p \in \mathcal{O}} M(Z_p; Z_q), \quad (3)$$

where the first term measures the similarity between the input exemplar  $\mathcal{I}$  and  $\mathcal{O}$  via our local neighborhoods metric as defined in Equation (2). Specifically, for each output sample  $q \in \mathcal{O}$ , we find the corresponding input sample  $p \in \mathcal{I}$  with the most similar neighborhood (according to Equation (2)), and sum their squared neighborhood differences. Our goal is to find an output  $\mathcal{O}$  with a low energy value. For our normal image resizing applications, we assume as null. Furthermore, we follow the EM-like methodology in [Kwatra et al. 2005] to optimize Equation 3 because of its high quality and generality with different boundary conditions.

We perform our synthesis process in multi-resolutions through an iterative optimization solver. For Equation 2, we use larger  $\omega$  in lower resolution to increase the spatial constraint. This scheme helps to preserve the global appearance during the synthesis process, then we decrease the  $\omega$  value in higher resolution to avoid the local texel repeat. In all our experiment, we use a 3-level pyramid and within each level, from lower to higher, we fix  $\omega = 0.65, 0.25, 0.1$ .

**Adaptive neighborhood matching** In each iteration, we search for the most similar input neighborhood for each output sample and assign the exemplar patch from the matched neighborhood to the output. This will gradually improve the synthesis quality. During the search step, exhaustively examining every input sample to minimize the energy value in Equation 2 can be computationally expensive. Previous works use K-means [Kwatra et al. 2005] or K-coherence [Han et al. 2006] to find an approximate nearest neighborhood (ANN). These strategies can efficiently accelerate the search process. However, when the texel diversity increases, the ANNs may not be accurate enough to improve the neighborhood quality, which will cause dissatisfied results (Figure 11). Therefore, we also search for the exact nearest neighborhoods by brute-force method over the exemplar image. Since the nearest neighborhoods are independent from each other, we implement our EM-based synthesis algorithm fully on GPU by implementing the search in a parallel framework, which will dramatically accelerate the search process. Specifically, in each thread, we calculate the similarity of two neighborhoods in the M-step and perform the average operation for each pixel in the E-step. Moreover, to further accelerate the neighborhood matching process, we use an adaptive scheme to narrow the searching domain in finer layers. Since our synthesis algorithm is a hierarchical framework which contains three layers, we use an adaptive scheme to gradually narrow the searching domain. In layer 1 where the images are processed in the lowest resolution, we search the best patch from the whole exemplar for each patch in the resulting image, we search for the best matching from the whole exemplar. Then, in layer 2, for each patch in the resulting image, we narrow the searching domain to the 40% pixels of the exemplar around its corresponding patch. Furthermore, we narrow the searching domain to 20% in the finest layer. Note that in the two finer layers, we still perform full search in the first matching operation and narrow the domain in the latter steps.



**Figure 11:** We perform accurate similar neighborhood search during synthesis process. The original image is shown in Figure 6(a).



**Synthesis as a whole** We synthesize the image as a whole when most contents of the scene are textures (usually more than 70%). The advantage of this strategy is that it can better preserves the global visual appearance and effectively reduce the object broken artifacts. Figure 2(b), 10(b), and 20(b) show 3 typical examples of this class. They are directly generated by our synthesis operator.

### 3.6 Merge of T-Regions and NT-Regions

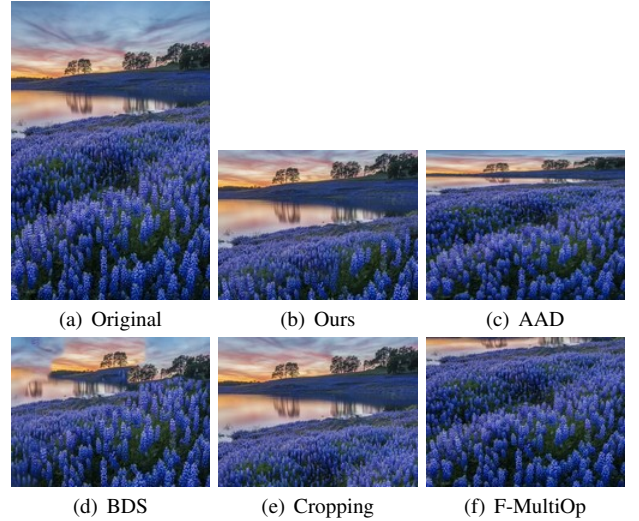
Since we resize the T- and NT- regions by different strategy, there may exist discontinuity of image contents between them. As demonstrated in Figure 12(b), the image content on the boundary between T- and NT-regions may be changed after synthesis. To reduce the discontinuity artifact, we grow the boundary by expanding 4 pixels on both inward and outward sides. We then get an overlapping area between the T- and NT-regions in the resized image. Afterwards, we re-synthesize those boundaries pixels by using the original image as the input example. The inclusion of NT- pixels on the boundary in the synthesis process helps to maintain the content consistency. Figure 12(b) and 12(c) compare the results without and with hiding the discontinuity, respectively.

## 4 Results and Discussion

We have implemented our method on a PC with Intel Core(TM) i7 950 CPU, 3.06 GHz, 8GB RAM, and nVidia Geforce GTX 760 GPU with 2048MB video memory. Our T-region synthesis algorithm is fully implemented on GPU with CUDA. The texture detection and saliency detection are both performed in real-time. The timing of resizing examples shown in this paper ranges between 10 seconds to 30 seconds, depending on the sizes of the output T-regions.

Figures 1, 2, 6-9, and 12-22 show our image retargeting results. We perform a user study for visual comparison (detailedly described below). For each figure, we put our result and primarily put other results with relatively higher votes. We can see that our content-aware synthesis method can preserve the overall texture features in terms of texel shape, perspective, content completeness, and clarity. The perspective appearance remains perspective. The shapes of texels are reasonably preserved, without over-squeezing/over-stretching or uneven distortion of texels within the regions. All the prominent contents of textures appear in the result.

**Evaluation on Textural Scene Retargeting Dataset** Although images from RetargetMe benchmark [Rubinstein et al. 2010] have a large variety in their content, textural regions are primarily simple and smooth. To represent more general situations that real world images fall into, we construct a Textural Scene Retargeting Dataset (TSRD) with 52 images. They all contain diversified and large textural patterns (occupied more than 50% areas of the whole image). These images are collected from RetargetMe (6 images), CSSD [Yan et al. 2013b] (Figure 8(a)) and internet. Some images in the RetargetMe benchmark which also contain textures are not included in TSRD because either the T-regions are small or the textures are relatively smooth without obvious texels (such as a still water surface, a smooth snowfield, and a manicured lawn). The images in the new dataset can be roughly divided into three types: pure textures with vivid global visual effects (Type 1), images with textures around one or more salient objects (Type 2), images with distinct T- and NT-regions (Type 3). For the exemplars in this paper, Figures 7(a), 14(a), 8(a) and 9(a) belong to Type 1. Figures 2(a), 10(a), 15(a) and 16(a) belong to Type 2. Figures 1(a), 5(a), 6(a), 10(a), 12(a), 13(a), 17(a), 18(a), 19(a) and 20(a) belong to Type 3. The whole dataset and the comparisons with previous state-of-the-arts methods are all shown in the supplemental materials.

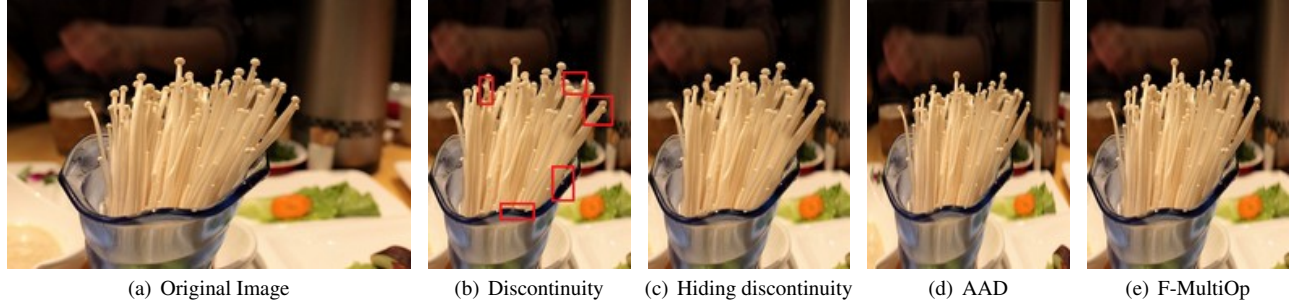


**Figure 13: Visual comparison.** Our method better preserves the scene layout and the perspective effect of the lavender. Perspective is lost in the cropping result. 67.27% users favour our result.

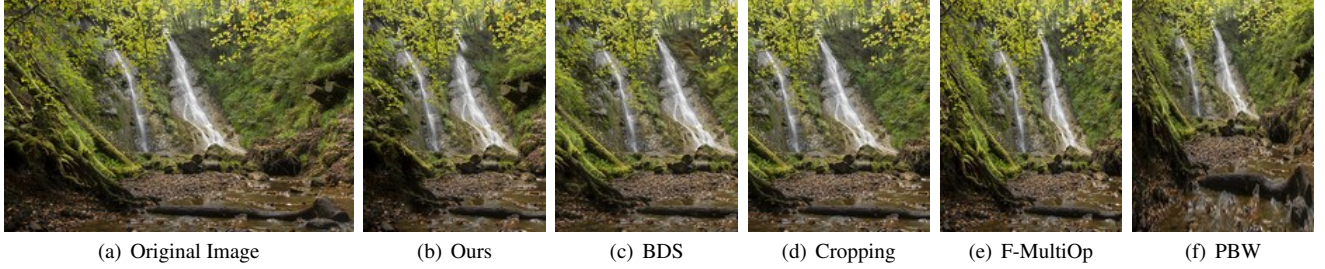
**Comparison with previous methods** For quantitative evaluation, we compare our method with six state-of-the-art image retargeting approaches, i.e., Axis-Aligned Deformation (AAD) [Panozzo et al. 2012], Bi-Directional Similarity (BDS) [Simakov et al. 2008], Cropping, Multi-Operator (F-MultiOp [Dong et al. 2012] and MultiOp [Rubinstein et al. 2009]), Patch-Based Warping (PBW) [Lin et al. 2013] and Shift-Map [Pritch et al. 2009]. The experiments are performed on our data set.

For AAD and PBW, we choose them for comparison since they are two typical continuous image warping approaches, which have been recently presented and testified to be among the best warping methods. SV [Krähenbühl et al. 2009] is also a good warping method which has been proved by the test on RetargetMe benchmark. However, in [Panozzo et al. 2012] the user study demonstrates that AAD is better than SV, so we only compare our method with AAD and PBW. The AAD results are generated with authors' program by using the default parameters. The PBW results are provided by the original author. When dealing with images in TSRD, compared to our method, the main problem of AAD and PBW is in many cases they will over-squeeze some contents (e.g., Figures 1(e), 13(c) and 20(c)) or the salient objects (e.g., Figures 15(e) and 19(f)) while over-stretch the background (Figure 16(f)), which makes some visually important regions to be too small in the resulting images. In many results the content structures of the scenes are obvious imbalance. The main reason is because warping usually tends to maintain as many as contents while preserving the aspect ratios of the areas with large energy or significance values. In most images of TSRD, these areas are usually the T-regions (Type 3) or the salient objects (Type 2). Therefore, to maintain the shape of those "important" areas, we can find that in the results generated by AAD or PBW, the T-regions are either overstretched (e.g., Figure 16(f)) or over-squeezed (e.g., Figure 20(c)). Uneven distortion to the salient objects may also appear if their significance values are low, such as Figures 2(c) and 18(c). Specifically, as shown in Figure 14(f), when the scene is almost all constructed by textures (Type 1), all the contents maybe be distorted if we use warping-based methods.

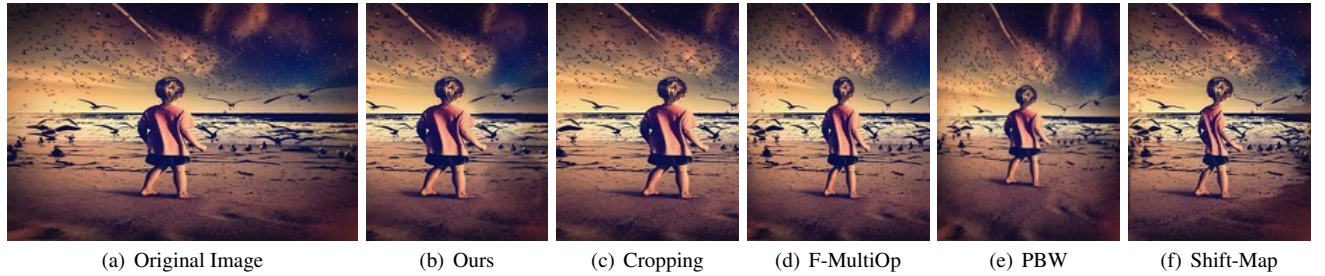
For BDS, we choose it for comparison since it is a synthesis-based image summarization method. The results are generated by imagestack program (<http://code.google.com/p/imagestack/>). For each exemplar, we use different parameters



**Figure 12:** Reducing the discontinuity artifact by growing the T-region boundaries in the original image, so as to provide overlapping regions for determining the seamless cut-path in the result. Input resolution  $500 \times 333$ , target resolution  $250 \times 333$ . 63.64% users favour our result.



**Figure 14:** Input resolution  $500 \times 333$ . Target resolution  $260 \times 333$ . 50.91% users favour our result.



**Figure 15:** Input resolution  $500 \times 340$ . Target resolution  $260 \times 340$ . 43.64% users favour our result.



**Figure 16:** Input resolution  $500 \times 331$ . Target resolution  $250 \times 331$ . 61.82% users favour our result.

to generate four images and manually choose the best one as the final result. When dealing with the images in TSRD, compared to our method, the main problem of BDS is that there will be obvious boundary discontinuity, such as the tire in Figure 2(d), the sky in Figure 13(d), the bottom-left lawn in Figure 19(c), and the grass-land in Figure 18(d). The reason is that BDS only uses color distance for neighbourhood matching, while the integration of spatial information in our algorithm can ensure the content continuity. Another artifact usually appears in BDS is the over-smoothing some areas, for example the tulips in Figure 1(c). We consider that it is due to the strategy of bidirectional similarity, sometimes on area in the resulting image is "obliged" to be similar as multiple areas of the original image. Our single-directional framework can avoid this problem. In fact, for image retargeting application, content loss is

allowed. Most of the users will be satisfied if the important contents are preserved. On the other hand, missing a good significance map also causes the loss of visually important contents in the results, such as the missing of stumps in Figure 14(c) and salient red flowers in Figure 17(c). Our good saliency map also makes it enough for our optimization process to only use a single-directional neighbourhood matching since the important areas are preserved in the initial retargeting operation. This also efficiently accelerates the speed of the synthesis process. In our experiments, we find that BDS usually costs about 5 minutes to generate an image, which limits its practical use in many applications.

*For Cropping*, we choose it for comparison since in most cases it is the first choice of the users during the comparative study of [Rubinstein et al. 2010]. On the other hand, a texture usually appears a

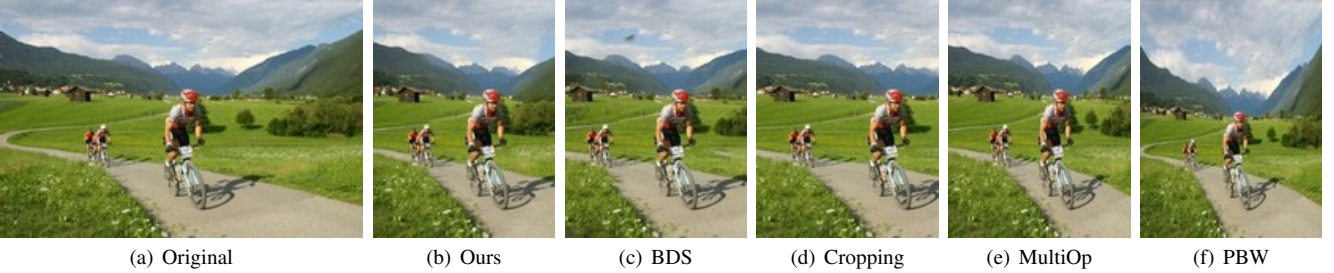




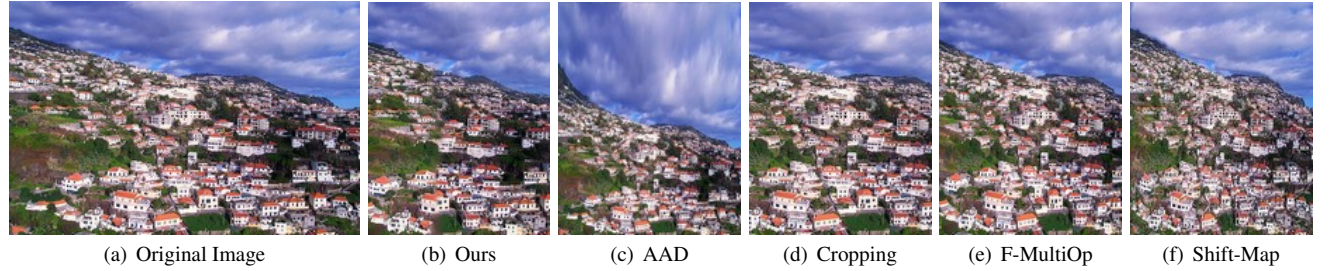
**Figure 17:** Input resolution  $400 \times 400$ . Target resolution  $200 \times 400$ . 69.09% users favour our result.



**Figure 18:** Input resolution  $500 \times 333$ . Target resolution  $260 \times 333$ . 52.73% users favour our result.



**Figure 19:** Input resolution  $460 \times 300$ . Target resolution  $230 \times 300$ . 43.64% users favour our result.



**Figure 20:** Input resolution  $500 \times 333$ . Target resolution  $260 \times 333$ . 52.73% users favour our result.

certain self-similarity, so maybe a simple cropping will be enough to well summarize its content. The results are created by an expert photographer. When dealing with images in TSRD, compared to our method, the main problem of cropping is some important contents will be unavoidably lost if there are multiple important contents located near the different sides of the input image, such as the vines and the stumps in Figure 14(d), the bush in Figure 16(d), three red flowers in Figure 17(d), the fence in Figure 14(d) and the trees and mountain in Figure 19(d). Our synthesis strategy can narrow the distance between the important contents and make them to appeared together in the result. Moreover, as discussed in [Rubinstein et al. 2010], cropping should be considered as a reference, not as

a proper retargeting algorithm. Here we still decide to compare with cropping only because sometimes it can benefit from the self-similarity characteristic of some textures and generate good retargeting results.

For *F-MultiOp* and *MultiOp*, we choose them for comparison since the *MultiOp* framework outperforms most algorithms according to the comparative study [Rubinstein et al. 2010]. *F-MultiOp* method has been demonstrated in [Dong et al. 2012] that it can generate results of the similar quality as *MultiOp*, so we consider these two methods as the same in our comparison. The *MultiOp* results of the six images collected from RetargetMe benchmark

are directly downloaded from the AAD website (<http://ig1.ethz.ch/projects/retargeting/aa-retargeting/aa-comparisons/dataset/index.html>), including the AAD results of those six images. The other results are generated by using F-MultiOp, which are all provided by the original author. When dealing with images in TSRD, compared to our method, the main problem of multi-operator methods is the even distortion to objects or texels, such as the tulips in Figure 1(d), the tire in Figure 2(f), the girl and flowers in 16(e), the flowers in 17(e), and the sportsman in Figure 19(e). The main reason is because although the integration of cropping operator can somewhat avoid the overall distortion, the unavoidable use of seam carving and homogeneous scaling operators (to protect the similarity between original image and resulting image) may still cause uneven distortions to objects or texels, especially when the T-regions are distributed throughout one dimension of the original image, such as Figures 1(a) and 17(a). This problem can only be solved by using a synthesis-based strategy. On the other hand, as shown in Figure 13(f), some important contents may be over-squeezed due to the lack of a good significance map.

For *Shift-Map*, we choose it for comparison since sometimes it can generate a synthesis-like result which selectively stitches some contents together to form a resized image. The results are partly provided by the original author, partly generated with the authors' online system, and partly generated with a public implementation (<https://code.google.com/p/retarget-toolkit/>) after the online system is taken down. When dealing with images in TSRD, compared to our method, the main problem of Shift-Map is that in many cases it will squeeze the content too much to cause uneven distortion (e.g., Figure 15(f) and 18(f)) or degenerate to cropping which will cause unexpected content loss (e.g., Figure 1(f) and 17(f)). The main reason is because stitching are minimized due to the global smoothness term. On the other hand, to get a good retargeting result by using Shift-Map, sometimes the user need to gradually resize the image by manually setting the number of removed columns/rows. This strategy is useful in preserving salient objects in the resulting image but ineffective for our images because for textures it usually does not contain distinct long boundaries that can help to penalize the removal of a large area. We consider that this is just the reason that in some cases shift-map degenerates to cropping when dealing with our images.

For *example-based texture synthesis*, apparently the normal texture synthesis algorithms such as texture optimization [Kwatra et al. 2005] and appearance-space texture synthesis [Lefebvre and Hoppe 2006] are not fit for image retargeting since they are originally designed for enlargement but have no effective schemes for size decrease. Inverse texture synthesis (ITS) [Wei et al. 2008] can produce a small texture compaction that summarizes the original. Its framework is very similar as BDS method so it will suffer the same problems as BDS if being used for image retargeting. On the other hand, the textural contents in most our images are not standard textures so using pure texture synthesis framework will easily cause content discontinuity or damage the globally varying effects.

**User Study** To evaluate our method further, we perform a user study to compare the results from different methods. All the stimuli are shown in the supplemental material. A total of 55 participants (24 males, 21 females, age range 20-45) from different backgrounds attended the comparison of 52 sets of resized images. Each participant is paid \$10 for their participation. All the participants sat in front of a 22-inch computers of  $1680 \times 1050$  px in a semi-dark room. In the experiment, we showed the original image, our result, and the images of the competitors. We then ask which image the participant prefers. For each group, the original image is separately shown in the first row, while the results are randomly displayed in

two additional rows within the same page. We allow the participant to choose at most two favourite images from the results. We did not provide a time constraint for the decision time. However, we recommend for the participants to finish the tests within 30 min. We allow the participants to move back and forth across the different pages by clicking the mouse. The average finishing time is 26 min 45 sec. A total of 4122 votes are reported. Figure 21 shows the statistics of how many times the results of each method has been chosen as favourite retargeting results. Based on the statistics, our method outperforms all competitors in general. For each test exemplar in TSRD, we show the percentages when our method and the competitors have been chosen by the participants in the supplemental material.

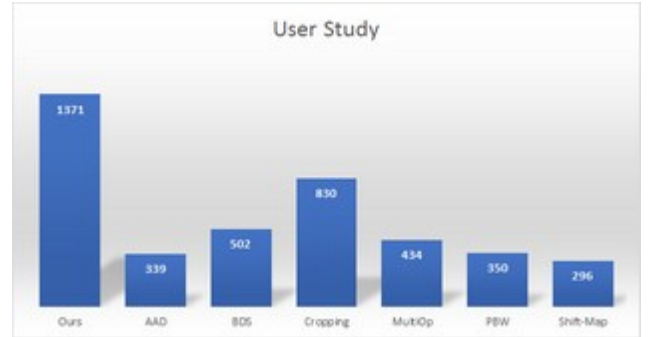


Figure 21: The statistics of the user study result.

**Limitations** The main limitation of our algorithm is the speed. Although we implement our synthesis operator fully on GPU, we still cannot get real-time performance like most warping-based methods, especially when the T-regions are large. Our method may generate unsatisfied results when the texels are very large (like an object) and have different attributes (color, shape, orientation, etc.). Figure 22 shows such one example, the texels (a candy) are large and visually different from each other. Therefore, we can see that there are obvious object discontinuity in our result.

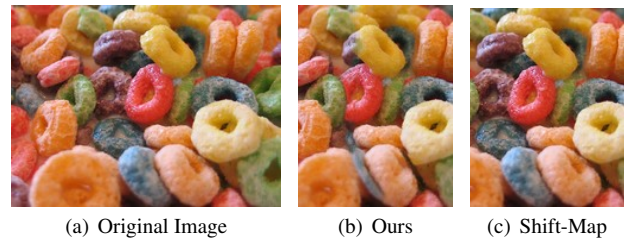


Figure 22: Input resolution  $456 \times 340$ . Target resolution  $264 \times 340$ . 29.09% users favour our result.

## 5 Conclusion and Future Work

The scenes containing textural regions are very common in natural images. However, as shown in our paper, most of them cannot be well handled by current general image resizing algorithms due to the lack of high level semantic information. We introduces a novel concept and robust method to solve the problem. An automatic methodology is proposed to detect the textures and adjust the saliency information. Then we use a synthesis-based image resizing system to achieve natural resizing effects with minimum texel



visual appearance damage. The integration of the spatial information ensures the content consistency between the original image and the result images. Our spatial-aware strategy can be integrated into most existing general resizing frameworks and enhance their robustness. Experiments shown that our system can handle a great variety of input scenes especially non-standard textural regions (for example Figure 10(a) is combined with many separate textural objects). For future work, extending the example-based synthesis operator to 3D scene resizing can be an interesting direction.

## References

- ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SÜSSTRUNK, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 11, 2274–2282.
- AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. *ACM Trans. Graph.* 26, 3, 10:1–10:10.
- BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 3, 24:1–24:12.
- BASHA, T., MOSES, Y., AND AVIDAN, S. 2013. Stereo seam carving a geometrically consistent approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 10, 2513–2525.
- BERGMAN, R., NACHLIELI, H., AND RUCKENSTEIN, G. 2007. Detection of textured areas in images using a disorganization indicator based on component counts. Tech. Rep. HPL-2005-175R1, HP Labs.
- BUADES, A., LE, T. M., MOREL, J.-M., AND VESE, L. A. 2010. Fast cartoon + texture image filters. *IEEE Transactions on Image Processing* 19, 8, 1978–1986.
- CHANG, C.-H. 2012. A line-structure-preserving approach to image resizing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Washington, DC, USA, 1075–1082.
- CHEN, Q., LI, D., AND TANG, C.-K. 2013. Knn matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 9, 2175–2188.
- CHENG, M.-M., ZHANG, G.-X., MITRA, N., HUANG, X., AND HU, S.-M. 2011. Global contrast based salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 409–416.
- CHO, T. S., BUTMAN, M., AVIDAN, S., AND FREEMAN, W. T. 2008. The patch transform and its applications to image editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.
- CRIMINISI, A., SHARP, T., ROTHER, C., AND P’EREZ, P. 2010. Geodesic image and video editing. *ACM Trans. Graph.* 29, 5 (Nov.), 134:1–134:15.
- DONG, W., ZHOU, N., AND PAUL, J.-C. 2008. Perspective-aware texture analysis and synthesis. *Vis. Comput.* 24, 7, 515–523.
- DONG, W., ZHOU, N., PAUL, J.-C., AND ZHANG, X. 2009. Optimized image resizing using seam carving and scaling. *ACM Trans. Graph.* 28, 5, 125:1–125:10.
- DONG, W., BAO, G., ZHANG, X., AND PAUL, J.-C. 2012. Fast multi-operator image resizing and evaluation. *Journal of Computer Science and Technology* 27, 1, 121–134.
- DONG, W., ZHOU, N., LEE, T.-Y., WU, F., KONG, Y., AND ZHANG, X. 2014. Summarization-based image resizing by intelligent object carving. *IEEE Transactions on Visualization and Computer Graphics* 20, 1, 111–124.
- GAL, R., SORKINE, O., AND COHEN-OR, D. 2006. Feature-aware texturing. In *Proc. of Eurographics Symp. on Rendering*, 297–303.
- GOFERMAN, S., ZELNIK-MANOR, L., AND TAL, A. 2012. Context-aware saliency detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 10 (Oct.), 1915–1926.
- GUO, Y., LIU, F., SHI, J., ZHOU, Z.-H., AND GLEICHER, M. 2009. Image retargeting using mesh parametrization. *IEEE Trans. Multi.* 11, 5, 856–867.
- HAN, J., ZHOU, K., WEI, L.-Y., GONG, M., BAO, H., ZHANG, X., AND GUO, B. 2006. Fast example-based surface texture synthesis via discrete optimization. *The Visual Computer* 22, 918–925.
- ILEA, D. E., AND WHELAN, P. F. 2011. Image segmentation based on the integration of colour-texture descriptors-a review. *Pattern Recogn.* 44, 10-11, 2479–2501.
- KARACAN, L., ERDEM, E., AND ERDEM, A. 2013. Structure-preserving image smoothing via region covariances. *ACM Trans. Graph.* 32, 6 (Nov.), 176:1–176:11.
- KAUFMANN, P., WANG, O., SORKINE-HORNUNG, A., SORKINE-HORNUNG, O., SMOLIC, A., AND GROSS, M. 2013. Finite element image warping. *Computer Graphics Forum* 32, 2pt1, 31–39.
- KIM, W., AND KIM, C. 2011. A texture-aware salient edge model for image retargeting. *Signal Processing Letters, IEEE* 18, 11, 631–634.
- KRÄHENBÜHL, P., LANG, M., HORNUNG, A., AND GROSS, M. 2009. A system for retargeting of streaming video. *ACM Trans. Graph.* 28, 5, 126:1–126:10.
- KWATRA, V., ESSA, I., BOBICK, A., AND KWATRA, N. 2005. Texture optimization for example-based synthesis. *ACM Trans. Graph.* 24, 3, 795–802.
- LEFEBVRE, S., AND HOPPE, H. 2006. Appearance-space texture synthesis. *ACM Trans. Graph.* 25, 3 (July), 541–548.
- LIN, S.-S., YEH, I.-C., LIN, C.-H., AND LEE, T.-Y. 2013. Patch-based image warping for content-aware retargeting. *IEEE Transactions on Multimedia* 15, 2, 359–368.
- MARGOLIN, R., TAL, A., AND ZELNIK-MANOR, L. 2013. What makes a patch distinct? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1139–1146.
- PANG, W.-M., CHOI, K.-S., AND QIN, J. 2013. Fast Gabor texture feature extraction with separable filters using gpu. *Journal of Real-Time Image Processing*, 1–9.
- PANOZZO, D., WEBER, O., AND SORKINE, O. 2012. Robust image retargeting via axis-aligned deformation. *Computer Graphics Forum* 31, 2.
- PERAZZI, F., KRAHENBUHL, P., PRITCH, Y., AND HORNUNG, A. 2012. Saliency filters: Contrast based filtering for salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 733–740.

- PRITCH, Y., KAV-VENAKI, E., AND PELEG, S. 2009. Shift-map image editing. In *IEEE International Conference on Computer Vision (ICCV)*, 151–158.
- RAO, A., AND LOHSE, G. 1993. Identifying high level features of texture perception. *CVGIP: Graphical Models and Image Processing* 55, 3, 218 – 233.
- RUBINSTEIN, M., SHAMIR, A., AND AVIDAN, S. 2008. Improved seam carving for video retargeting. *ACM Trans. Graph.* 27, 3, 16:1–16:10.
- RUBINSTEIN, M., SHAMIR, A., AND AVIDAN, S. 2009. Multi-operator media retargeting. *ACM Trans. Graph.* 28, 3, 23:1–23:12.
- RUBINSTEIN, M., GUTIERREZ, D., SORKINE, O., AND SHAMIR, A. 2010. A comparative study of image retargeting. *ACM Trans. Graph.* 29, 6, 160:1–160:10.
- SIMAKOV, D., CASPI, Y., SHECHTMAN, E., AND IRANI, M. 2008. Summarizing visual data using bidirectional similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.
- TARGHI, A. T., BJÖRKMAN, M., HAYMAN, E., AND EKLUNDH, J.-O. 2006. Real-time texture detection using the lu-transform. In *ECCV Workshop on Computation Intensive Methods for Computer Vision*.
- TODOROVIC, S., AND AHUJA, N. 2009. Texel-based texture segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, 841 –848.
- WANG, Y.-S., TAI, C.-L., SORKINE, O., AND LEE, T.-Y. 2008. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.* 27, 5, 118:1–118:8.
- WEI, L.-Y., HAN, J., ZHOU, K., BAO, H., GUO, B., AND SHUM, H.-Y. 2008. Inverse texture synthesis. *ACM Trans. Graph.* 27, 3, 52:1–52:10.
- WEI, L.-Y., LEFEBVRE, S., KWATRA, V., AND TURK, G. 2009. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, Eurographics Association, 93–117.
- WOLF, L., GUTTMANN, M., AND COHEN-OR, D. 2007. Non-homogeneous content-driven video-retargeting. In *Proc. of the Eleventh IEEE Int. Conf. on Computer Vision (ICCV-07)*, 1–6.
- WU, H., WANG, Y.-S., FENG, K.-C., WONG, T.-T., LEE, T.-Y., AND HENG, P.-A. 2010. Resizing by symmetry-summarization. *ACM Trans. Graph.* 29, 6, 159:1–159:10.
- XU, L., YAN, Q., XIA, Y., AND JIA, J. 2012. Structure extraction from texture via relative total variation. *ACM Trans. Graph.* 31, 6 (Nov.), 139:1–139:10.
- YAN, J., LIN, S., KANG, S. B., AND TANG, X. 2013. Learning the change for automatic image cropping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 971–978.
- YAN, Q., XU, L., SHI, J., AND JIA, J. 2013. Hierarchical saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Washington, DC, USA, 1155–1162.
- YANG, C., ZHANG, L., LU, H., RUAN, X., AND YANG, M.-H. 2013. Saliency detection via graph-based manifold ranking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3166–3173.
- ZHANG, J., AND KUO, C.-C. 2012. Region-adaptive texture-aware image resizing. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 837–840.
- ZHANG, Y. F., HU, S. M., AND MARTIN, R. R. 2008. Shrinkability maps for content-aware video resizing. *Computer Graphics Forum* 27, 7, 1797–1804.
- ZHANG, L., SONG, M., ZHAO, Q., LIU, X., BU, J., AND CHEN, C. 2013. Probabilistic graphlet transfer for photo cropping. *IEEE Transactions on Image Processing* 22, 2, 802–815.
- ZHANG, L., SONG, M., YANG, Y., ZHAO, Q., ZHAO, C., AND SEBE, N. 2014. Weakly supervised photo cropping. *IEEE Transactions on Multimedia* 16, 1, 94–107.